

Das Handbuch zu bmeps 2.1.5

Dipl.-Ing. D. Krause

4. Januar 2010

Inhaltsverzeichnis

1	Übersicht	3
1.1	Einsatzzweck	3
1.2	Vorwort zur Version 2.0.0	3
1.3	Lizenzbedingungen	4
1.3.1	Allgemeine Lizenzbedingungen für bmeps	4
1.3.2	Ausnahmen	4
2	Installation	5
3	Nutzung	6
3.1	Konfigurations-Mechanismus	6
3.2	Kommandozeilen-Optionen	8
3.3	Konfigurationsdatei	10
3.4	Konfigurationseinträge	11
3.5	LaTeX und bmeps	22
3.5.1	Verfahrensweisen	22
3.5.2	Beispiele	23
4	Bitmap-Dateitypen	32
4.1	PNG	32
4.2	JPEG	32
4.3	NetPBM	33
4.4	TIFF	33
5	Neuerungen in Version 2.0.0	34
5.1	Robustere PS/EPS-Ausgabe	34
5.2	Multiple data sources	36
5.3	Passthrough von JPEG-Dateien	36
5.4	Flexible Bitbreite	37
5.5	PDF-Ausgabe	37

1 Übersicht

1.1 Einsatzzweck

Das Programm „bmeps“ wandelt PNG-, JPEG- und NETPBM-Dateien nach EPS und PDF um. Die Ausgabe von BoundingBox-Dateien, die von L^AT_EX benötigt werden, ist ebenfalls möglich.

TIFF-Dateien, die mit der Funktion *TIFFReadRGBAImage()* aus der libtiff-Bibliothek eingelesen werden können, werden ebenfalls verarbeitet.

Zusätzlich zum ausführbaren Programm bmeps wird beim Compilieren der Software eine Bibliothek libbmeps erstellt, die von dem Programm dvips genutzt werden kann.

1.2 Vorwort zur Version 2.0.0

Die Version bmeps 1.x.x funktionierte bereits recht gut, jedoch gingen mir seit längerer Zeit mögliche Verbesserungen durch den Kopf:

- Hinzunahme von PDF als Ausgabeformat. Bisher gab es zu diesem Zweck das Programm png2pdf, das jedoch nur PNG-Dateien verarbeiten konnte.
- Verbesserungen am Code waren angebracht, insbesondere hinsichtlich der Wartungsfreundlichkeit. Alle Aufgaben hinsichtlich Komprimierung und Encoding waren in einer Funktion gebündelt, das Hinzufügen neuer Kompressionsverfahren wäre nur sehr schwer möglich. Die Nutzung der dklibs-Bibliotheken vereinfacht die Kombination unterschiedlicher Kompressions- und Encodingverfahren ungemein.
Die Codeteile zum Einlesen der Bitmap-Dateien waren mit Codeteilen zur Ausgabe verwoben. Die jetzt vorgenommene Trennung der Ein- und Ausgabefunktionen vereinfacht es, Support für weitere Dateitypen hinzuzufügen.
- Teilweiser Support für TIFF-Dateien wurde hinzugefügt, hauptsächlich um TIFF-Dateien verarbeiten zu können, die von einem Fax-Gateway erzeugt wurden.
Zum Einlesen von TIFF-Dateien wird die Funktion *TIFFReadRGBAImage()* verwendet, das Programm bmeps kann also nur solche TIFF-Dateien verarbeiten, die von dieser Funktion unterstützt werden.
- Die PS-Ausgabe farbiger Bilder kann getrennte Datenquellen für Rot, Grün und Blau verwenden. Dies ermöglicht eine Run-Length-Komprimierung für farbige Flächen.

1.3 Lizenzbedingungen

1.3.1 Allgemeine Lizenzbedingungen für bmeps

Die Software ist unter den nachfolgenden Lizenzbedingungen im BSD-Stil an Sie lizenziert:

- Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 - Neither the name of the Dirk Krause nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
- This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

1.3.2 Ausnahmen

- Das Unterverzeichnis „dvips-mods“ der Sourcecode-Distribution enthält modifizierte dvips-Quellen. Für diese gelten weiterhin die Copyright- und Lizenzbestimmungen der originalen dvips-Quellen.

Die Originalquellen finden Sie in der teTeX-Distribution¹. Beim Auspacken entsteht ein Unterverzeichnis „tetex-src-2.0.2/texk/dvips“ (die Versionsnummer kann variieren), in den Quellen und Dokumentationen finden Sie genauere Informationen zu den Lizenzbedingungen.
- Die Datei „DOCU/kant_krishna/excel_to_eps.txt“ und der Inhalt des Verzeichnisses „contrib/kant_krishna“ wurden von Krishna Kant bereitgestellt, von ihm wurden weder Copyright- noch Lizenzbedingungen spezifiziert.

Ich empfehle, die Zeile „Author: . . .“ unverändert zu lassen, wenn diese Dateien weitergegeben bzw. modifiziert werden.

¹z.B. unter <ftp://ftp.dante.de/tex-archive/systems/unix/teTeX/current/distrib>

2 Installation

Zum Compilieren aus den Quelltexten werden folgende Bibliotheken benötigt:

- zlib (erforderlich)
- libbz2 (optional)
- libpng (erforderlich)
- jpeglib (erforderlich)
- NetPBM (optional)
- libtiff (optional)
- dklibs (erforderlich)

Nach das Archiv entpackt wurde, können die üblichen Kommandos

```
./configure  
make  
make install
```

verwendet werden.

3 Nutzung

3.1 Konfigurations-Mechanismus

Bmeps bietet eine Vielzahl von Konfigurationseinstellungen. Um diese nicht bei jedem Programmstart komplett anzugeben, kann eine Konfigurationsdatei `/etc/bmeps/bmeps.conf` bzw. `/usr/local/etc/bmeps/bmeps.conf` verwendet werden, in der verschiedene Konfigurationen definiert werden. Als Kommandozeilenoption muss dann nur mit der Option „-l“ eine der definierten Konfigurationen ausgewählt werden.

Mit der Option „-o *Name=Wert*“ können einzelne Einstellungen der gewählten Konfiguration überschrieben werden.

Wird keine „-l“-Option verwendet, ermittelt bmeps folgendermaßen eine Konfiguration (die Suche wird nach dem ersten erfolgreichen Test beendet):

- Wenn ein Name für die Ausgabedatei angegeben ist, wird getestet, ob dieser auf „eps“, „ps“, „pdf“ oder „bb“ endet. Falls ja, wird die Konfiguration „eps“, „ps“, „pdf“ bzw. „bb“ verwendet.

Hinweis: Wird keine Option „-l...“ angegeben, müssen alle „-o...“-Optionen in der Kommandozeile *nach* dem Namen der Ausgabedatei stehen!

- Wenn die Umgebungsvariable EPSOUTPUT existiert, enthält diese den Namen der zu nutzenden Konfiguration.

- Es wird in den permanenten Optionen (Präferenzen) nach einer mit

`bmeps -c -l ...`

vorgenommenen Standardeinstellung gesucht.

- Falls bisher keine Konfiguration gefunden wurde, wird „eps“ verwendet.

Eine Reihe von Konfigurationen ist bereits in bmeps einprogrammiert (siehe Tabelle 1), für diese Konfigurationen ist keine Konfigurationsdatei erforderlich.

Tabelle 1: Standard-Konfigurationen

Name	Bedeutung
ps1	Erzeugung von PS level 1
ps2	Erzeugung von PS level 2
ps3	Erzeugung von PS level 3
ps	Synonym für ps2
eps1	Erzeugung von EPS level 1
eps2	Erzeugung von EPS level 2
eps3	Erzeugung von EPS level 3
wird fortgesetzt	

<i>Fortsetzung</i>	
eps	Synonym für eps2
faxps	Erzeugung einer Druckdatei schwarz/weiß im A4-Format (PS level 2)
pdf12	Erzeugung von PDF 1.2
pdf13	Erzeugung von PDF 1.3
pdf14	Erzeugung von PDF 1.4
pdf	Synonym für pdf14
faxpdf	Erzeugung von PDF 1.4 schwarz/weiß, Seitenanpassung an A4-Format
bb	Erzeugung einer Bounding-Box-Datei für \LaTeX

3.2 Kommandozeilen-Optionen

Bmeps wird mit dem Kommando

`bmeps [Optionen] [Eingabedatei] [Ausgabedatei] [Optionen]`

oder

`bmeps [Optionen] [Verzeichnis]`

aufgerufen.

Folgende Optionen können verwendet werden:

- `-h`
zeigt einen Hilfetext an.
- `-v`
zeigt die Versionsnummer und die Lizenzbedingungen an.
- `-c -l...`
speichert die angegebene Konfiguration als permanente Option (Standardeinstellung) ab.
- `-u`
löscht alle permanenten Optionen.
- `-C`
zeigt die aktuelle Konfiguration an.
- `-lKonfiguration`
wählt die angegebene Konfiguration aus. Optional können – durch Kommata abgetrennt – einzelne Konfigurationseinstellungen überschrieben werden.
Die mit „-l“ vorgenommenen Einstellungen können mit „-c“ als permanente Optionen (Standardeinstellungen) abgespeichert werden.
- `-oName=Wert`
überschreibt einen Konfigurationseintrag (kann mehrfach verwendet werden).
- `-fFrame-Nummer`
`-fStart-Frame-End-Frame`
wählt die zu verarbeitenden Frames aus (bei Dateien, die mehrere Bilder in einer Datei enthalten wie z.B. TIFF).
- `-m`
`-m-`
aktiviert bzw. deaktiviert den make-Modus. Wird bmeps für ein Verzeichnis gestartet, wird das Verzeichnis nach Bitmap-Dateien durchsucht, die bmeps verarbeiten kann. Für gefundene Dateien wird eine Konvertierung gestartet.
Im make-Modus wird geprüft, ob bereits eine bei der Konvertierung entstehende Ausgabedatei vorhanden ist. Eine Konvertierung wird nur gestartet, wenn entweder noch keine

Ausgabedatei existiert oder die Ausgabedatei nicht aktuell ist.
Diese Einstellung kann mit „-c“ als permanente Option gespeichert werden.

- -a
erzeugt bei Bedarf automatisch einen Namen für die Ausgabedatei aus dem Namen der Eingabedatei und dem Ausgabetreiber.
Wenn kein Ausgabedateiname angegeben ist, schreibt bmeps normalerweise in die Standardausgabe. Mit „-a“ kann stattdessen der Name für die Ausgabedatei automatisch gebildet werden.
Mit
`bmeps -a -lbb picture.png`
kann die Datei picture.bb erzeugt werden.
- -t*Dateityp*
gibt den Dateityp an („png“ für PNG-Dateien, „jpg“ für JPEG-Dateien, „tif“ für TIFF-Dateien oder „pbm“ für NetPBM-Dateien). Diese Option ist nur erforderlich, wenn die Standardeingabe verarbeitet wird und demzufolge der Name der Quelldatei nicht bekannt ist.
- -A
-A-
unterdrückt bzw. erlaubt die Meldung, dass ein Alpha-Kanal von PNG bzw. TIFF nach PDF übernommen wurde.
Werden PDF-Bilder in *.tex-Dateien verwendet, ist eine spezielle Anweisung in der Dokumentenpräambel ratsam, auf diese wird in der Meldung hingewiesen.

Wird keine Option „-l...“ verwendet, versucht bmeps, aus dem Namen der Ausgabedatei die gewünschte Konfiguration zu ermitteln. In diesem Fall dürfen Optionen zum Überschreiben von Konfigurationseinträgen erst *nach* dem Namen für die Ausgabedatei stehen.

3.3 Aufbau der Konfigurationsdatei

Die Konfigurationsdatei kann ein oder mehrere Abschnitte enthalten, jeder Abschnitt beschreibt eine Konfiguration.

Jeder Abschnitt beginnt mit dem Konfigurations-Namen in eckigen Klammern, z.B.

```
[drucker]
```

Optional kann durch Doppelpunkt abgetrennt ein Name einer bereits definierten Konfiguration angegeben werden, deren Einstellungen geerbt werden, z.B.

```
[drucker:ps2]
```

Der Rest eines jeden Konfigurationsabschnittes besteht aus Zeilen, die jeweils einen Konfigurationseintrag enthalten, siehe Abschnitt 3.4 auf der nächsten Seite.

Ein Beispieleintrag für einen PS2-Schwarz-Weiß-Drucker könnte also folgendermaßen aussehen:

```
[ drucker : ps2 ]  
color = no
```

3.4 Konfigurationseinträge

Konfigurationseinträge können bmeps auf verschiedene Arten übergeben werden. Um beispielsweise eine Ausgabedatei im A4-Format für einen PS2-schwarz/weiß-Drucker zu erzeugen sind folgende Kommandos möglich:

```
bmeps -l ps2 -o color=no -o media.size=A4 ...
bmeps -l ps2 -o c=n -o m.s=A4
bmeps -l ps2 , color=no , media.size=A4 ...
bmeps -l ps2 , c=n , m.s=A4 ...
```

Alternativ dazu könnte auch eine Konfigurationsdatei mit dem Inhalt

```
[ printer : ps2 ]
color           =          no
media size      =          A4
```

erstellt werden und bmeps mit der Kommandozeile

```
bmeps -l printer ...
```

gestartet werden.

Für Konfigurationsdateien empfehle ich, die Namen der Konfigurationseinträge auszuschreiben. Die Abkürzungen sollten nur auf der Kommandozeile angewendet werden.

Folgende Konfigurationseinträge können gesetzt werden:

- output type

o.t

Ausgabeformat, entweder „ps“, „pdf“ oder „bb“.

```
eps1: ps
eps/eps2: ps
eps3: ps
ps1: ps
ps/ps2: ps
ps3: ps
pdf12: pdf
pdf13: pdf
pdf/pdf14: pdf
faxpdf: pdf
bb: bb
```

- level

l

Format-Level, entweder „1“, „2“ oder „3“ für PS oder „1.2“, „1.3“ oder „1.4“ für PDF.

```

    eps1: 1
  eps/eps2: 2
    eps3: 3
    ps1: 1
  ps/ps2: 2
    ps3: 3
    pdf12: 1.2
    pdf13: 1.3
  pdf/pdf14: 1.4
    faxpdf: 1.4

```

- `color`

`c`

Diese boolsche Option entscheidet für PDF und PS Level 2 und 3, ob die Ausgabe farbig oder in Graustufen erfolgt.

```

    eps1: off
  eps/eps2: on
    eps3: on
    ps1: off
  ps/ps2: on
    ps3: on
    pdf12: on
    pdf13: on
  pdf/pdf14: on
    faxpdf: off

```

- `pdf fit`

`p.f`

legt den Zoomfaktor beim Öffnen der PDF-Datei fest:

- `width`
so dass die Bildbreite die Fensterbreite ausnutzt,
- `height`
so dass die Bildhöhe die Fensterhöhe ausnutzt oder
- `page`
so dass das Bild komplett im Fenster dargestellt werden kann.

```

    pdf12: page
    pdf13: page
  pdf/pdf14: page
    faxpdf: width

```

- `interpolate`

`i`

setzt das Flag für Image-Interpolation in PDF und PS-Level 2 und 3.

eps/eps2: on
eps3: on
ps/ps2: on
ps3: on
pdf12: on
pdf13: on
pdf/pdf14: on
faxpdf: on

- jpeg interpolate

j.i

Beim Durchschleusen DCT-codierter JPEG-Dateien wird normalerweise die Image-Interpolation deaktiviert. Mit dieser boolschen Option kann die Image-Interpolation auch für DCT-codierte Daten aktiviert.

eps/eps2: off
eps3: off
ps/ps2: off
ps3: off
pdf12: off
pdf13: off
pdf/pdf14: off
faxpdf: off

- encoding

e

Encoding- und Komprimierungsverfahren. Die Verfahren werden als Liste angegeben, die einzelnen Verfahren werden durch Doppelpunkt getrennt, z.B. „ascii85:run-length:flate:dct“ bzw. „a:r:f:d“.

Möglich sind folgende Verfahren (siehe Abb. 1 auf der nächsten Seite für mögliche Kombinationen):

- ascii85
a

ASCII85-Encoding. Ist dieses Verfahren deaktiviert, wird ASCII-Hex-Encoding eingesetzt.

- run-length
r

Run-Length-Komprimierung.

- flate
f

Flate-Komprimierung

- dct
d

DCT-Komprimierung. Bmeps nimmt selbst keine DCT-Komprimierung vor, kann

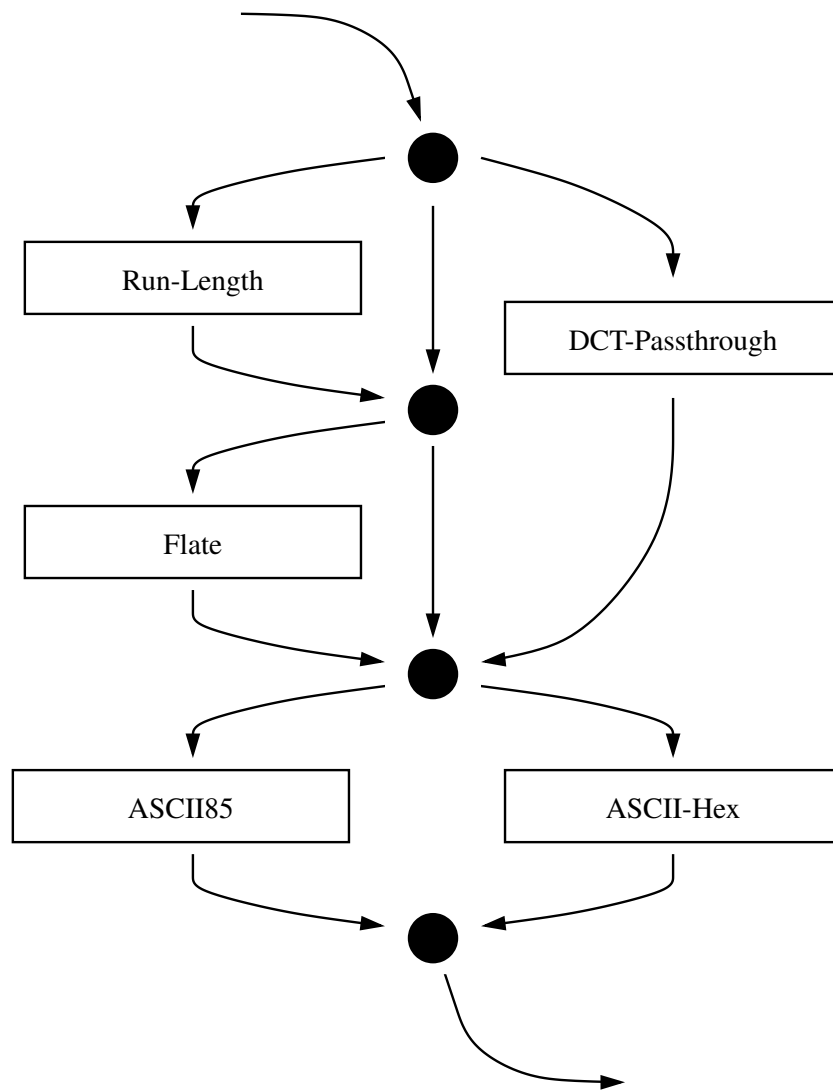


Abbildung 1: Komprimierungs- und Encodingverfahren

aber DCT-komprimierte JPEG-Dateien 1:1 zu Ausgabedateien umwandeln (mit ASCII-Hex- oder ASCII85-Encoding).

- eps1: ASCII-Hex
- eps/eps2: ASCII85, DCT
- eps3: ASCII85, flate, DCT
- ps1: ASCII-Hex
- ps/ps2: ASCII85, DCT
- ps3: ASCII85, flate, DCT
- pdf12: ASCII85, flate, DCT
- pdf13: ASCII85, flate, DCT
- pdf/pdf14: ASCII85, flate, DCT
- faxpdf: ASCII85, flate, DCT

- `jfif sof set`
`j.s.s`

Der PostScript-Standard erlaubt in DCT-codierten Daten nur die SOF-Marker SOF₀ und SOF₁. Je nach verwendetem Drucker bzw. PS-Interpreter können jedoch auch weitere SOF-Marker verarbeitet werden.

Die Zahlencodes der erlaubten Marker können als Liste angegeben werden, die einzelnen Codes werden durch Doppelpunkt getrennt, z.B. „0:1:2“ um SOF₀, SOF₁ und SOF₂ zu erlauben.

```
eps/eps2: 0:1
eps3: 0:1
ps/ps2: 0:1
ps3: 0:1
pdf12: 0:1
pdf13: 0:1
pdf/pdf14: 0:1
faxpdf: 0:1
```

- `predictor = string`

Beim Erstellen von PDF-Dateien kann ein Predictor genutzt werden, um die Kompressionsrate der flate-Kompression zu verbessern. Dabei werden nicht die Farbwerte aller Punkte gespeichert sondern die Differenz des aktuellen Farbwertes zu einem vorhergesagtem Wert. Der vorhergesagte Wert wird – je nach verwendetem Predictor – aus dem entsprechenden Wert bzw. Byte des Punktes links, oberhalb und/oder links/oberhalb des aktuellen Punktes ermittelt.

Es können die Predictoren „none“ (keinen Predictor verwenden, dies ist die Standardeinstellung), „tiff“ (TIFF-Sub-Predictor), „png-sub“, „png-up“, „png-average“ bzw. „png-paeth“ verwendet werden.

Für „normale“ PNG-Bilder (z. B. Screenshots, Logos, Diagramme. . .) wird kein Predictor benötigt, nur für Sonderfälle (z. B. Bilder mit Farbgradienten, als PNG gespeicherte Fotos. . .) kann mit einem Predictor eine bessere Komprimierung erzielt werden.

Falls nach dem Konvertieren einer PNG-Datei die PDF-Datei signifikant größer (deutlich mehr als 25 % Größenzuwachs) ist, sollte die Verwendung eines Predictors in Erwägung gezogen werden. Die Auswahl des Predictors muss dabei an das Bild angepasst werden, ggf. mehrere bzw. alle Predictoren durchprobieren, um eine möglichst kleine Ausgabedatei zu erhalten.

Der „tiff“-Predictor kann für beliebige Farbtiefen verwendet werden, „png-*“ nur bei 8 oder 16 Bits pro Farbkomponente.

RFC 2083 empfiehlt in Abschnitt 9.6 den „png-paeth“-Predictor, falls eine Anwendung zum Schreiben von PNG-Dateien nur einen fest eingebauten Predictor (in der RFC „Filter“ genannt) implementiert, da dieser im Mittel die besten Ergebnisse bringt. Dieser Predictor ist möglicherweise auch bei der Konvertierung zu PDF eine gute Wahl.

- `transfer alpha channel`
`t.a.c`

legt fest, ob die Alpha-Kanal-Daten aus der Eingabedatei in die Ausgabedatei übertra-

gen werden.

Dies ist nur möglich, wenn PDF-1.4-Ausgabe erzeugt wird.

Alpha-Datentransfer deaktiviert das Mischen gegen eine Hintergrundfarbe und die Erstellung von Image-Masken.

pdf/pdf14: on

faxpdf: on

- allow pdf page attributes

a.p.p.a

erlaubt bzw. verbietet die Erzeugung von PDF-Seitenattributen.

PDF-Seitenattribute werden benötigt, wenn ein Alpha-Kanal in die PDF-Ausgabe übertragen wird und die PDF-Datei als eigenständige Datei betrachtet wird. Ohne die PDF-Seitenattribute kann es zu einer fehlerhaften Farbdarstellung kommen.

- mix

m

legt fest, ob eine Farbmischung gegen eine Hintergrundfarbe erfolgen soll, wenn die Eingabedatei einen Alpha-Kanal enthält. Gemischt wird entweder gegen die Hintergrundfarbe aus dem Background-Chunk der Eingabedatei (falls vorhanden), gegen die Standard-Hintergrundfarbe oder gegen weiß.

eps1: on

eps/eps2: on

eps3: on

ps1: on

ps/ps2: on

ps3: on

pdf12: on

pdf13: on

pdf/pdf14: off

faxpdf: off

- background

b

gibt die Standard-Hintergrundfarbe vor, gegen die gemischt wird, wenn die Eingabedatei einen Alpha-Kanal aber keinen Background-Chunk enthält.

Die Farbangabe erfolgt als *rot:grün:blau*-Tripel mit Werten im Bereich von 0.0 bis 1.0.

```

    eps1:  weiß
eps/eps2: weiß
    eps3:  weiß
    ps1:   weiß
ps/ps2:   weiß
    ps3:   weiß
    pdf12: weiß
    pdf13: weiß
pdf/pdf14: weiß
    faxpdf: weiß

```

- `always use default background`

`a.u.d.b`

gibt an, ob die Standard-Hintergrundfarbe auch dann genutzt werden soll, wenn die Eingabedatei einen Background-Chunk enthält.

Als Werte sind hier „true“/„yes“/„on“ bzw. „false“/„no“/„off“ möglich.

```

    eps1:  off
eps/eps2: off
    eps3:  off
    ps1:   off
ps/ps2:   off
    ps3:   off
    pdf12: off
    pdf13: off
pdf/pdf14: off
    faxpdf: off

```

- `create image mask`

`c.i.m`

erzeugt eine Image-Maske, falls die Eingabedatei einen Alpha-Kanal enthält.

```

    eps3:  off
    ps3:   off
    pdf12: off
    pdf13: off
pdf/pdf14: off
    faxpdf: off

```

- `image mask trigger level`

`i.m.t.l`

gibt das Trigger-Level für die Erstellung der Image-Maske an. Die Angabe erfolgt als Wert im Bereich von 0.0 bis 1.0.

eps3: 0,000001
ps3: 0,000001
pdf12: 0,000001
pdf13: 0,000001
pdf/pdf14: 0,000001
faxpdf: 0,000001

- use resolution chunk

u.r.c

Falls diese boolsche Option gesetzt ist, werden die Daten zur Bildauflösung aus der Eingabedatei verwendet (falls vorhanden), um die Bildgröße festzulegen.

eps1: off
eps/eps2: off
eps3: off
ps1: off
ps/ps2: off
ps3: off
pdf12: off
pdf13: off
pdf/pdf14: off
faxpdf: off

- media size

m.s

gibt das Ausgabeformat vor. Die Angabe kann auf folgende Arten erfolgen:

- Format-Name: A3, A4, A5, B4, B5, Letter, Legal, Tabloid, Ledger, Statement, Executive, Folio, Quarto oder 10x14.
- 4 durch Doppelpunkte getrennte Zahlen x0, y0, x1 und y1, diese geben den linken unteren und den rechten oberen Punkt in PS-Einheiten an (72 PS-Einheiten sind ein Zoll, d.h. 2,54 cm).
- 8 durch Doppelpunkte getrennte Zahlen bx0, by0, bx1, by1, ix0, iy0, ix1 und iy1. Die Angaben bx0...by1 werden als Bounding-Box benutzt, das eigentliche Bild wird im Bereich ix0...iy1 platziert.

Hinweis: Die Optionen „u.r.c“ und „m.s“ können nicht gleichzeitig verwendet werden.

```

    eps1: -
  eps/eps2: -
    eps3: -
    ps1: -
  ps/ps2: -
    ps3: -
    pdf12: -
    pdf13: -
  pdf/pdf14: -
    faxpdf: A4
    bb: -

```

- `separated dictionary`
`s.d`

Diese boolsche Option legt für PS Level 2 und 3 fest, ob ein eigenes Dictionary genutzt wird.

```

  eps/eps2: on
    eps3: on
  ps/ps2: on
    ps3: on

```

- `vmreclaim`
`v`

Diese boolsche Option entscheidet für PS Level 2 und 3, ob am Ende der Ausgabe ein „1 vmreclaim“ eingefügt wird um eine Garbage Collection vorzuschlagen. Diese Option erfordert ein eigenes Dictionary (siehe `separated dictionary`).

```

  eps/eps2: off
    eps3: off
  ps/ps2: off
    ps3: off

```

- `operator dictionary`
`o.d`

Diese boolsche Option gibt für PS Level 2 und 3 vor, ob der ein `image`-Operator mit einem Dictionary verwendet wird oder ein `image/colorimage`-Operator mit mehreren einzelnen Argumenten.

```

  eps/eps2: on
    eps3: on
  ps/ps2: on
    ps3: on

```

- `multiple data sources`
`m.d.s`

Diese boolsche Option legt fest, ob getrennte Datenquellen für Rot, Grün und Blau erzeugt werden. Dies ermöglicht eine Run-Length-Komprimierung farbiger Flächen bzw. waagerechter Linien in der Ausgabedatei.

Voraussetzung ist allerdings, dass die gepackten Bits für eine Farbe einer Ausgabezeile in einem String untergebracht werden können, der nicht größer als 16383 Bytes ist.

eps/eps2: on
eps3: off
ps/ps2: on
ps3: off

- `showpage`

`sh`

Diese boolsche Option legt fest, ob die PS-Ausgabe einen `showpage`-Operator enthält. Der `showpage`-Operator sollte für Standalone-Bilder verwendet werden, die einzeln betrachtet oder gedruckt werden. Dateien, die in andere Dokumente eingebunden werden (z.B. mit \LaTeX) sollten keinen `showpage`-Operator enthalten.

Werden mehrere Frames aus der Eingabedatei konvertiert, wird der `showpage`-Operator immer mit benutzt, unabhängig von dieser Einstellung.

eps1: off
eps/eps2: off
eps3: off
ps1: on
ps/ps2: on
ps3: on

- `dsc comments`

`dsc.c`

Diese boolsche Option legt fest, ob DSC-Kommentare in PS-Ausgaben eingebaut werden.

eps1: on
eps/eps2: on
eps3: on
ps1: off
ps/ps2: off
ps3: off

- `draft`

`d`

Diese boolsche Option veranlasst, dass nur ein Platzhalter-Bild erzeugt wird. Hierzu wird aus der Eingabedatei nur die Bildgröße gelesen, als Ausgabe wird ein teilweise gefülltes Viereck mit diagonalen Streifen erzeugt.

Standardeinstellung: aus.

3.5 L^AT_EX und bmeps

3.5.1 Verfahrensweisen

Bitmap-Images können auf verschieden Art mit L^AT_EX verwendet werden:

- Die Bilddateien werden nach EPS umgewandelt, die EPS-Dateien werden durch die L^AT_EX-Quelle referenziert. Vorteil hierbei ist, dass die Konvertierung nur einmal erfolgen muss. Nachteilig ist der erhöhte Platzbedarf, da jedes Bild einmal als Originaldatei, einmal als separate EPS-Datei und einmal in der von dvips produzierten Ausgabedatei vorliegt.
- Die L^AT_EX-Quelle referenziert die Bitmap-Datei, beim Lauf von dvips wird dann die EPS-Version automatisch „on the fly“ produziert. Vorteil hierbei ist der geringere Platzbedarf.

Nachteilig ist der höhere Zeitaufwand, da die Konvertierung bei jedem dvips-Lauf erfolgt. Dieser Zeitaufwand kann minimiert werden, indem für Probedrucke der Draft-Modus von bmeps aktiviert wird.

Ein weiterer möglicher Nachteil ist, dass für alle Bilder dieselben Konvertierungseinstellungen verwendet werden.

Zum Lauf von L^AT_EX muss für jedes Bild eine Bounding-Box-Datei vorliegen.

Das Erzeugen der Daten „on the fly“ ist nur für EPS-Ausgabe zur Nutzung mit L^AT_EX/dvips möglich, nicht für PDF-Ausgabe zur Nutzung mit pdfL^AT_EX.

Ein typisches Nutzungsszenario beinhaltet folgende Schritte:

1. Einstellung einer Standardkonfiguration.
Wollen Sie beispielsweise PS2-Dateien für einen Schwarz-Weiß-Drucker erstellen, verwenden Sie das Kommando

```
bmeps -c -lps2 ,c=n
```
2. Bounding-Box-Dateien anlegen.
Um für alle Bitmap-Dateien im aktuellen Verzeichnis die entsprechenden Bounding-Box-Dateien anzulegen, können Sie das Kommando

```
bmeps -lbb .
```

verwenden.
3. L^AT_EX-Präambel bearbeiten.
Setzen Sie in die L^AT_EX-Präambel die Zeilen

```
\DeclareGraphicsRule{.png}{eps}{.bb}{'bmeps #1}  
\DeclareGraphicsRule{.jpg}{eps}{.bb}{'bmeps #1}  
\DeclareGraphicsRule{.jpeg}{eps}{.bb}{'bmeps #1}
```
4. Bitmap in der L^AT_EX-Quelle referenzieren.
Verwenden Sie

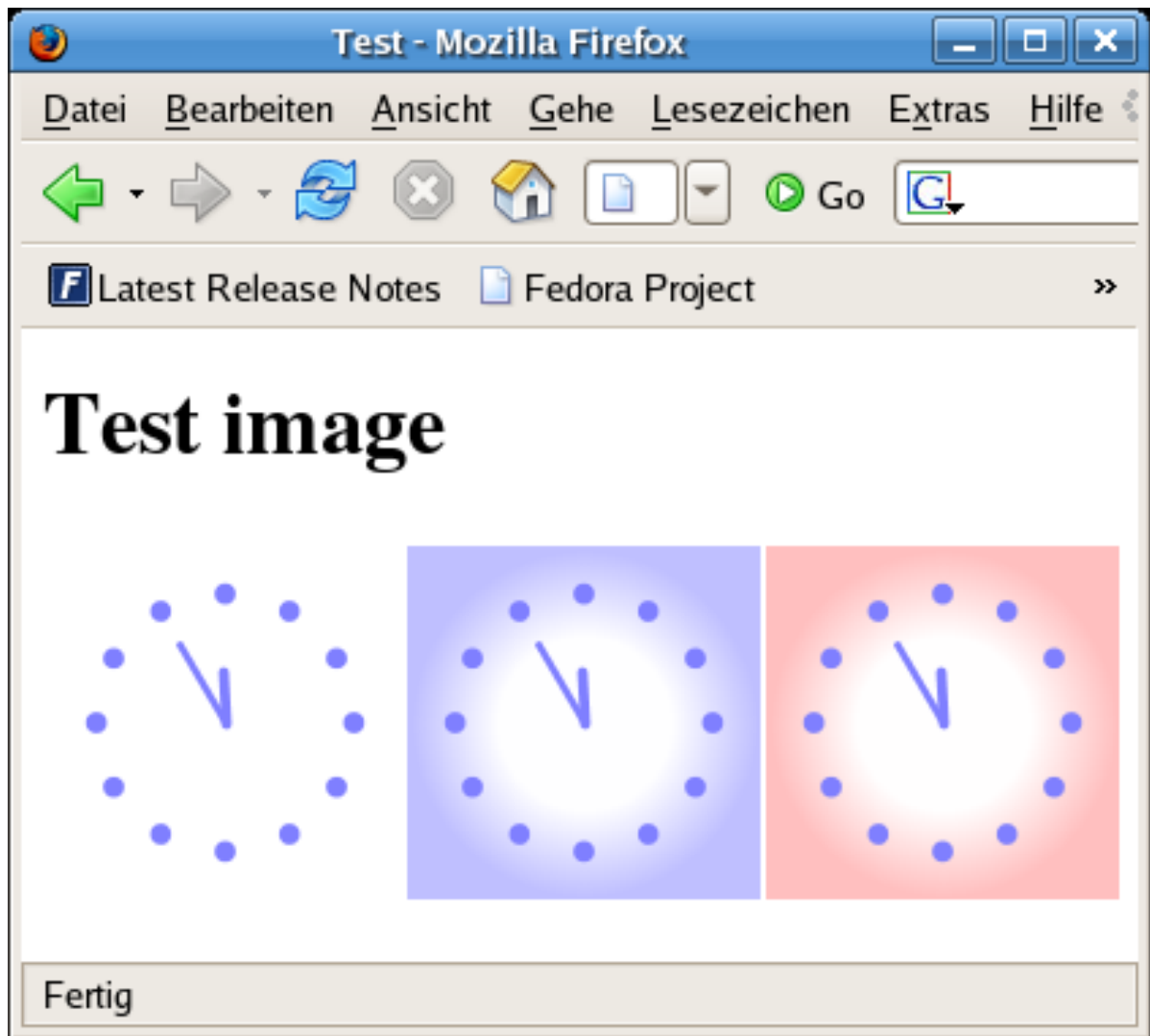


Abbildung 2: Beispiel-PNG vor verschiedenen Hintergründen

```
\includegraphics{input.png}
```

um das Bild „input.png“ in das Dokument einzufügen.

3.5.2 Beispiele

Test-Bild Für Demonstrationszwecke wird hier eine selbst erstellte PNG-Datei fbt.png verwendet. Diese zeigt eine auf 5 Minuten vor 12 stehende Uhr (five before twelve) mit Alpha-Kanal. Abb. 2 zeigt das Bild vor verschiedenen Hintergründen. Deutlich sichtbar ist das unterschiedlich starke Durchscheinen des Hintergrundes in verschiedenen Bildbereichen.

L^AT_EX-Dokument Zu Demonstrationszwecken wird hier eine Beamer-Präsentation mit nur einer Folie benutzt:

```
\documentclass{beamer}
\mode<presentation>{\usetheme{Madrid}}
\setbeamercovered{transparent}}
\usepackage[german]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{ifpdf}
\usepackage{graphicx}
\usepackage{color}
\ifpdf\hypersetup{pdfpagemode=FullScreen}\fi
\title[Beamer and bmeps]{Using bmeps with the beamer class}
\author[Krause]{D.~Krause}
\subject{bmeps}
\begin{document}
% \beamertemplateshadingbackground{yellow!50}{blue!50}
\begin{frame}
\frametitle{Image over colored background}
\includegraphics[width=5cm]{fbt1}
\end{frame}
\end{document}
```

Diese Präsentation hat zunächst einen einfarbigen weißen Hintergrund, im Lauf der Beispiele wird zu einfarbig gelben Hintergrund bzw. Farbverlauf von blau nach gelb gewechselt. Gleichzeitig ändert sich der Dateiname fbt1, da verschiedene Versionen des Bildes verwendet werden.

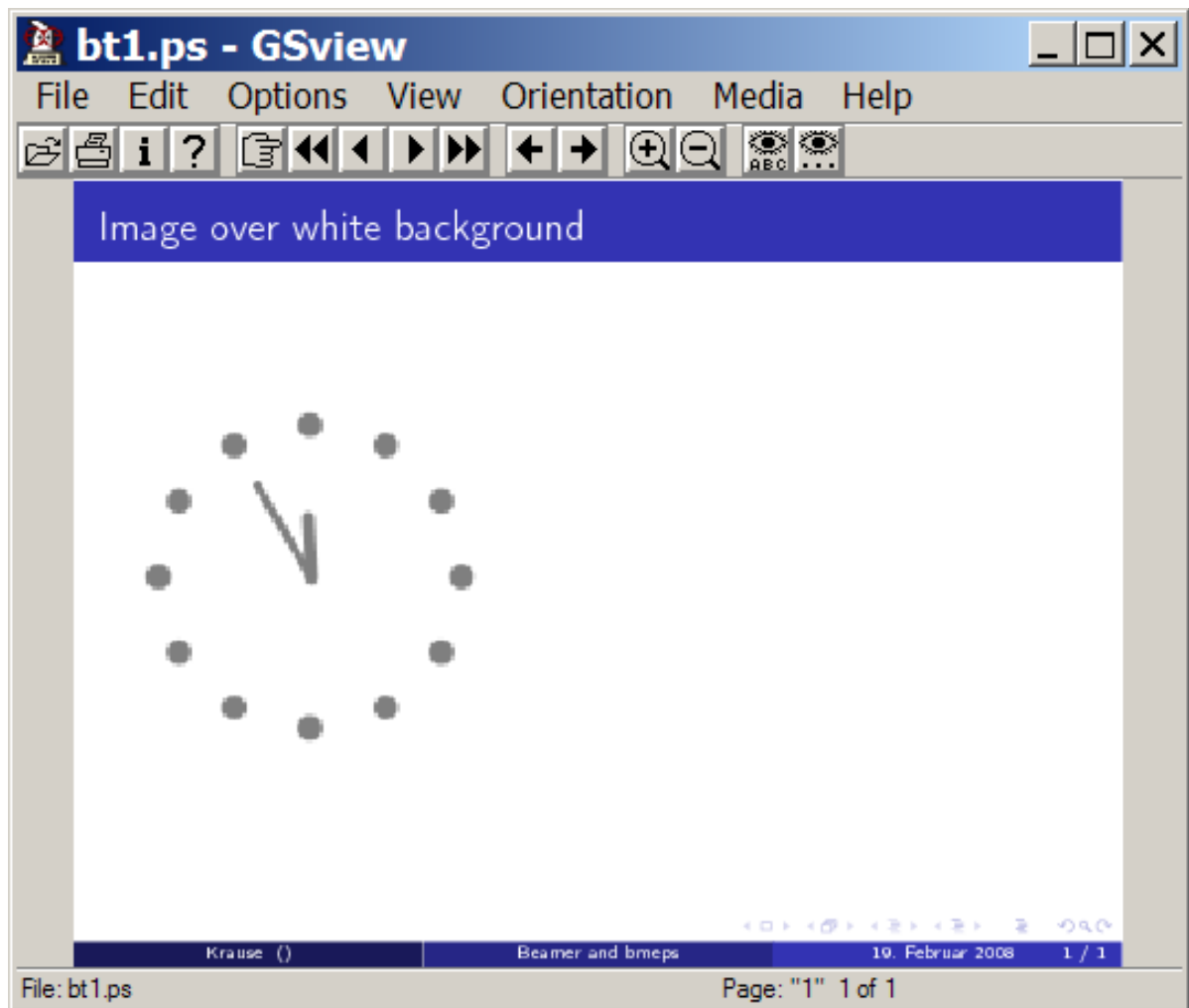


Abbildung 3: Beispiel 1

PS-Level 1 Wird die PNG-Datei mit
`bmps -lept1 fbt.png fbt1.eps`
nach EPS umgewandelt und die Präsentation mit
`latex bt1 && latex bt1 && dvips bt1`
erstellt, erhält man ein Ergebnis entsprechend Abb. 3.

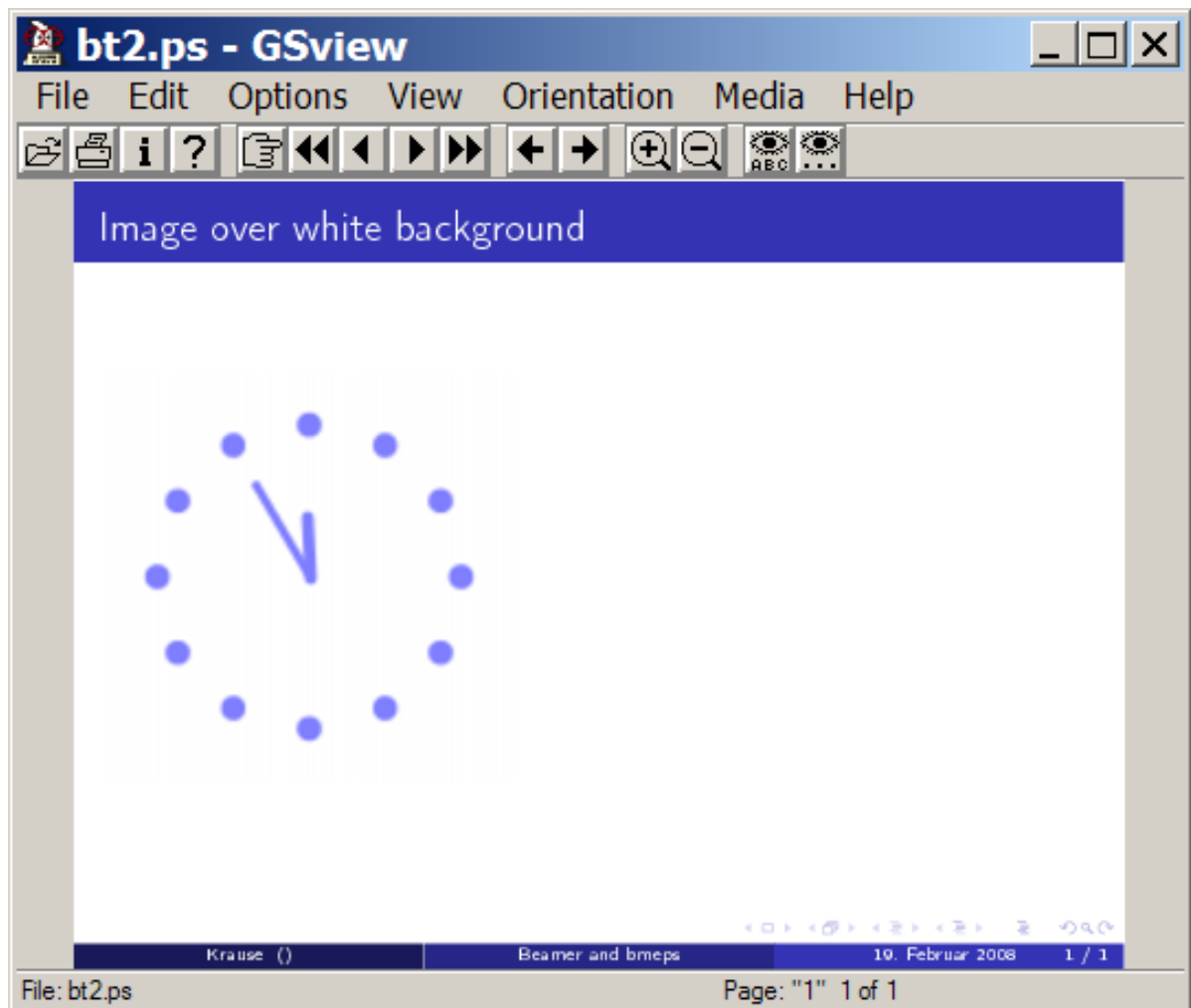


Abbildung 4: Beispiel 2

PS-Level 2 Die Umwandlung mit

```
bmps -lps2 fbt.png fbt2.eps
```

und das Einsetzen des resultierenden Bildes in die Präsentation erzeugen eine farbige Version des Bildes, siehe Abb. 4.

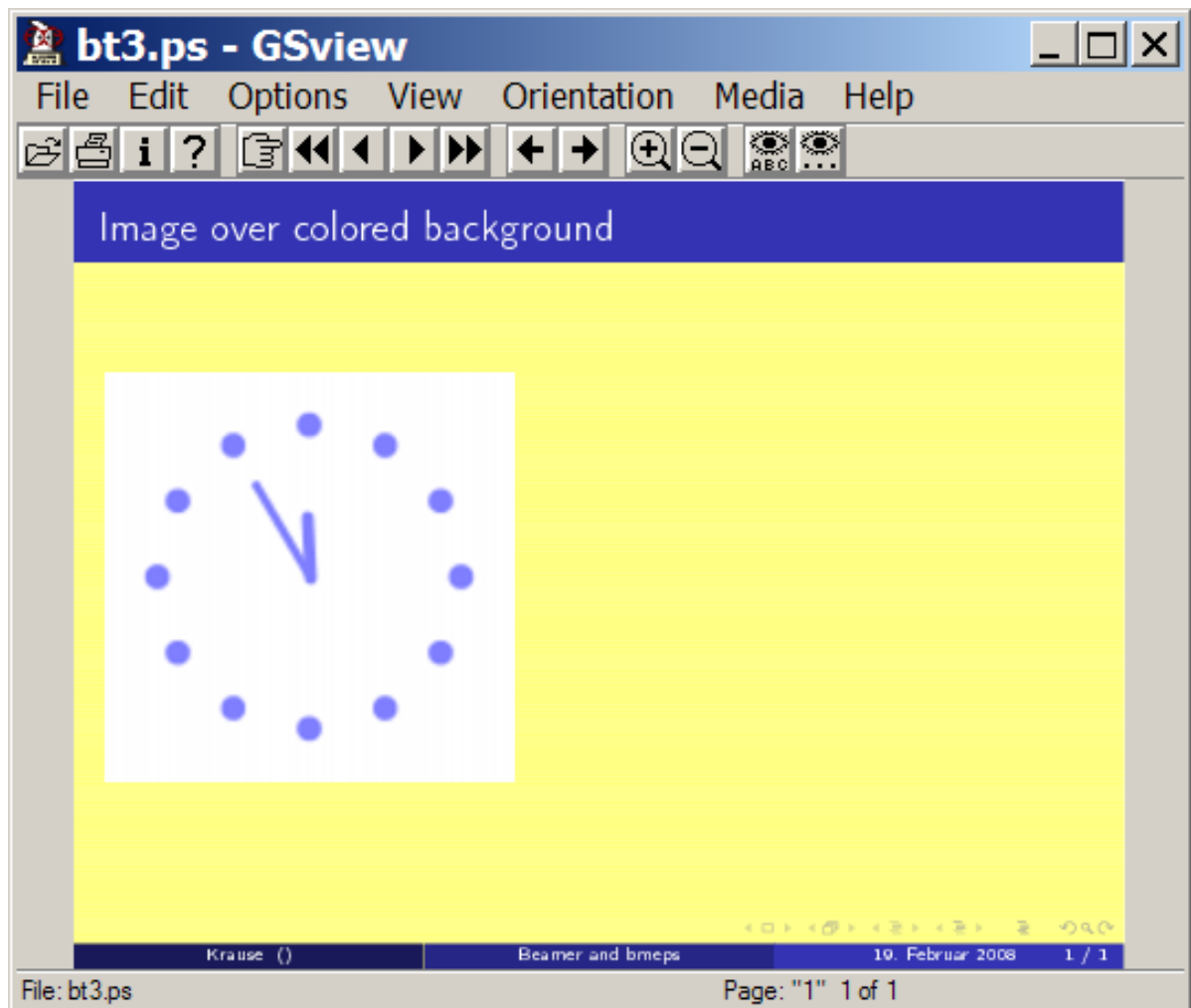


Abbildung 5: Beispiel 3

Farbiger Hintergrund Wird ein farbiger Hintergrund in der Präsentation gewählt, erscheint das eingebundene Bild nach wie vor auf weißem Hintergrund, da die weißen Bildpunkte in der EPS-Datei stehen, siehe Abb. 5. Dies kann aber korrigiert werden.

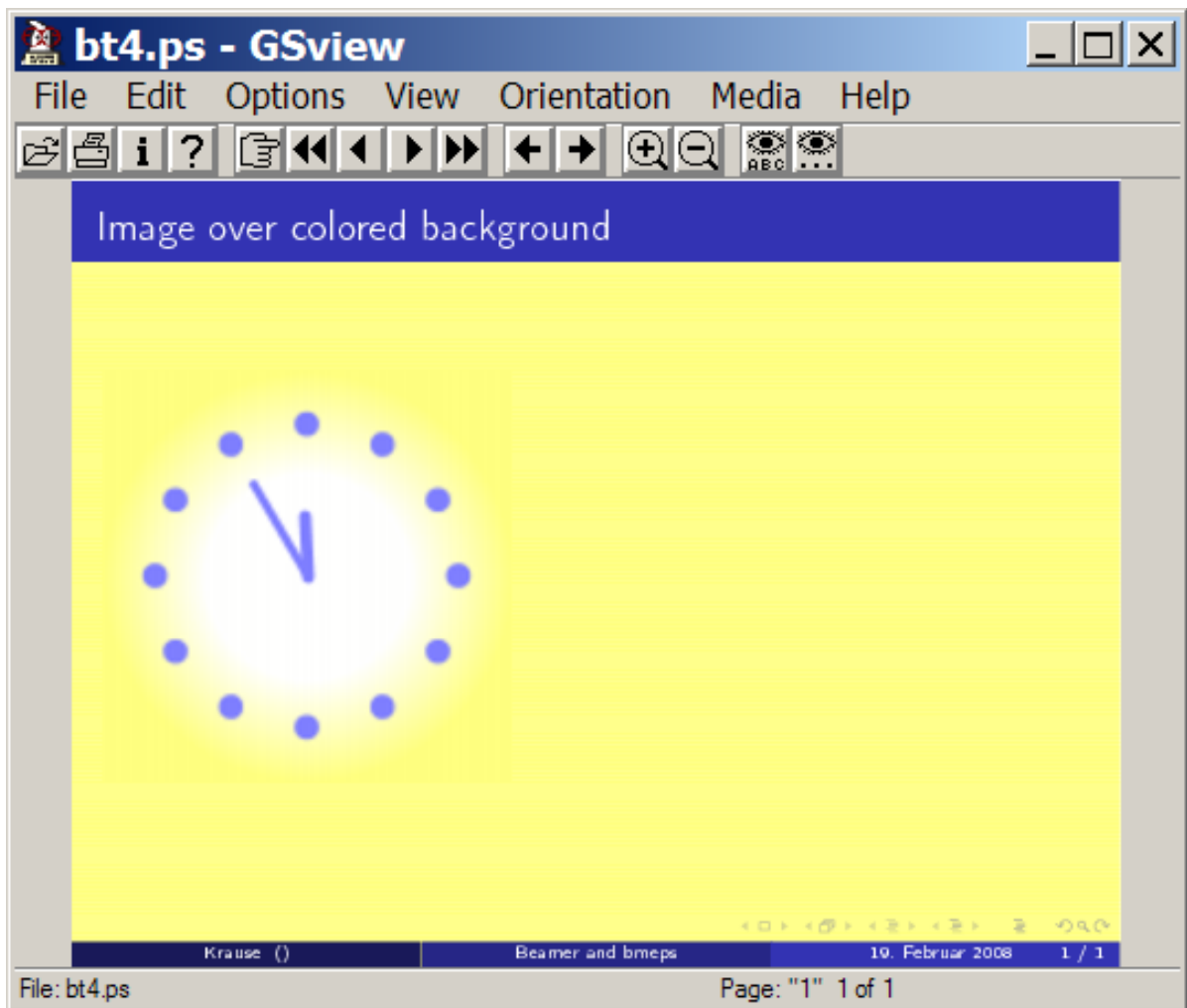


Abbildung 6: Beispiel 4

Anpassung an Dokumenten-Hintergrundfarbe Das Dokument (die Präsentation) verwendet einen gelben Hintergrund, als RGB-Werte treten 0xFF, 0xFF, 0x7F auf. Eine Umrechnung auf das Intervall [0;1] ergibt 1, 1, 0.5. Die Konvertierung erfolgt daher mit dem Kommando

```
bmeps -lps2 ,m=y ,b=1.0:1.0:0.5 ,a.u.d.b=y fbt.png fbt4.eps
```

Es wird gemischt (m=y), die Standardhintergrundfarbe ist 1.0:1.0:0.5, die Standardhintergrundfarbe wird auch dann verwendet, wenn die PNG-Datei Informationen zur Hintergrundfarbe enthält (a.u.d.b: always use default background).

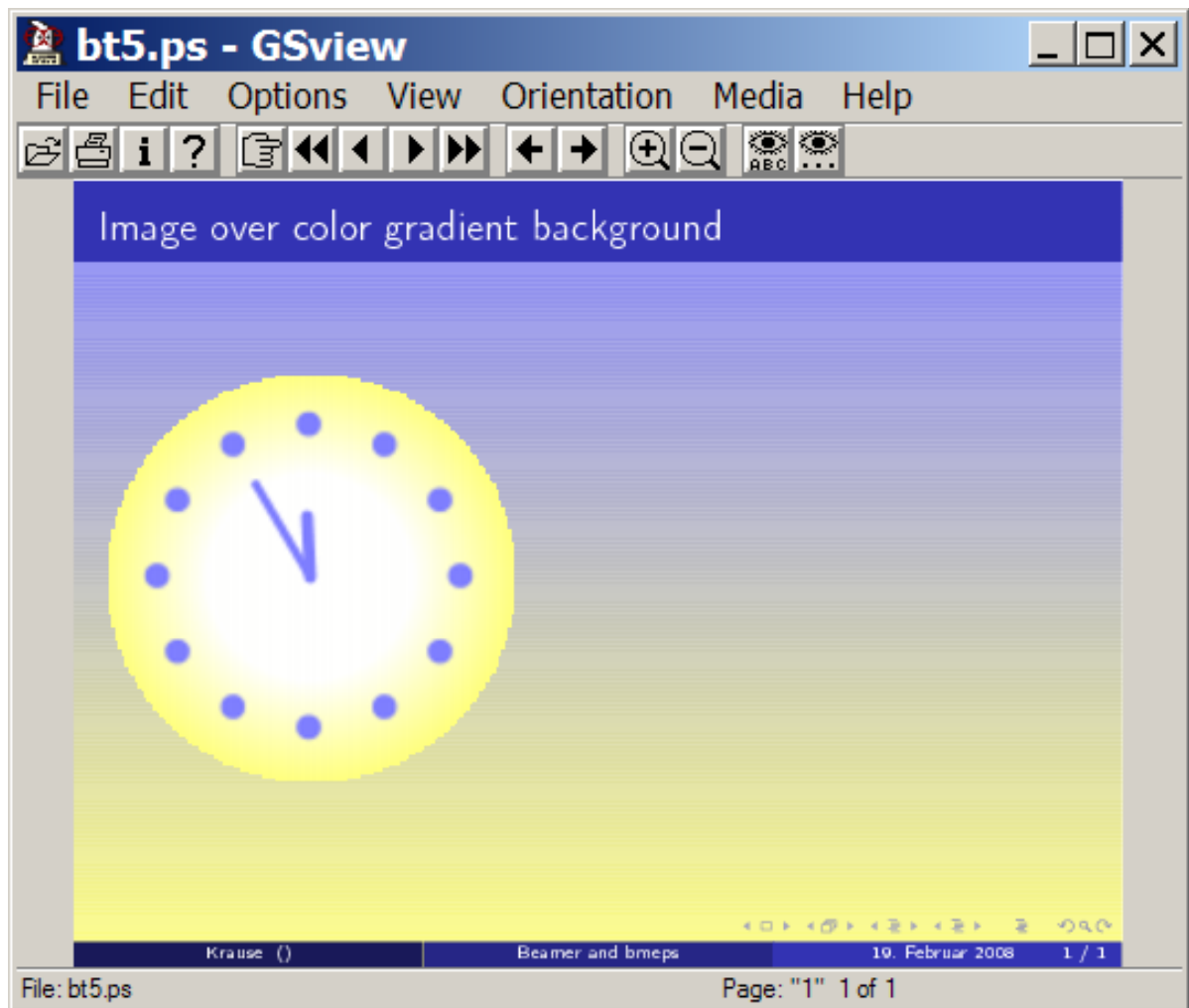


Abbildung 7: Beispiel 5

Maskierung (PS-Level 3) Im Beispiel 5 (Abb. 7) verwendet die Präsentation als Hintergrund einen Farbverlauf von blau nach gelb. Um kein sich deutlich abhebendes gelbes Quadrat zu erhalten, wird bei der Image-Umwandlung zusätzlich zum Mischen gegen gelben Hintergrund eine Image-Mask angelegt (c.i.m: create image mask). Dies bedeutet, es werden nur die Pixel des Images gezeichnet, deren Deckungskraft größer als 0 ist.

Das Kommando hierzu lautet:

```
bmeps -lps3 ,m=y ,b=1.0:1.0:0.5 ,a.u.d.b=y ,c.i.m=y fbt.png fbt5.eps
```

Alpha-Kanal-Transfer (PDF) Soll der Hintergrund durch das Image hindurchscheinen (siehe Abb. 8 auf der nächsten Seite), so muss der Alpha-Kanal mit in das umgewandelte Image übertragen werden. Dies ist nur für PDF-Ausgabe möglich. Die Konfiguration „pdf“ ist ein Synonym für „pdf14“, die Übertragung des Alpha-Kanales ist hier bereits voreingestellt.

```
bmeps -lpdf fbt.png fbt7.pdf
pdflatex bt7 && pdflatex bt7 && pdflatex bt7
```

Werden von einer *.tex-Datei PDF-Objekte mit Alpha-Kanal eingebunden, muss eine zusätzliche Zeile in die Präambel eingefügt werden:

```
\documentclass{beamer}
\mode<presentation>{\usetheme{Madrid}}
\setbeamercovered{transparent}}
\usepackage[german]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{ifpdf}
\usepackage{graphicx}
\usepackage{color}
\title[Beamer and bmeps]{Using bmeps with the beamer class}
\author[Krause]{D.~Krause}
\subject{bmeps}
% ——— The next line is important when dealing with alpha channels
\pdfpageattr{/Group << /S /Transparency /I true /CS /DeviceRGB>>}
\begin{document}
\beamertemplateshadingbackground{yellow!50}{blue!50}
\begin{frame}
\frametitle{Image over color gradient background}
\includegraphics[width=5cm]{fbt7.pdf}
\end{frame}
\end{document}
```

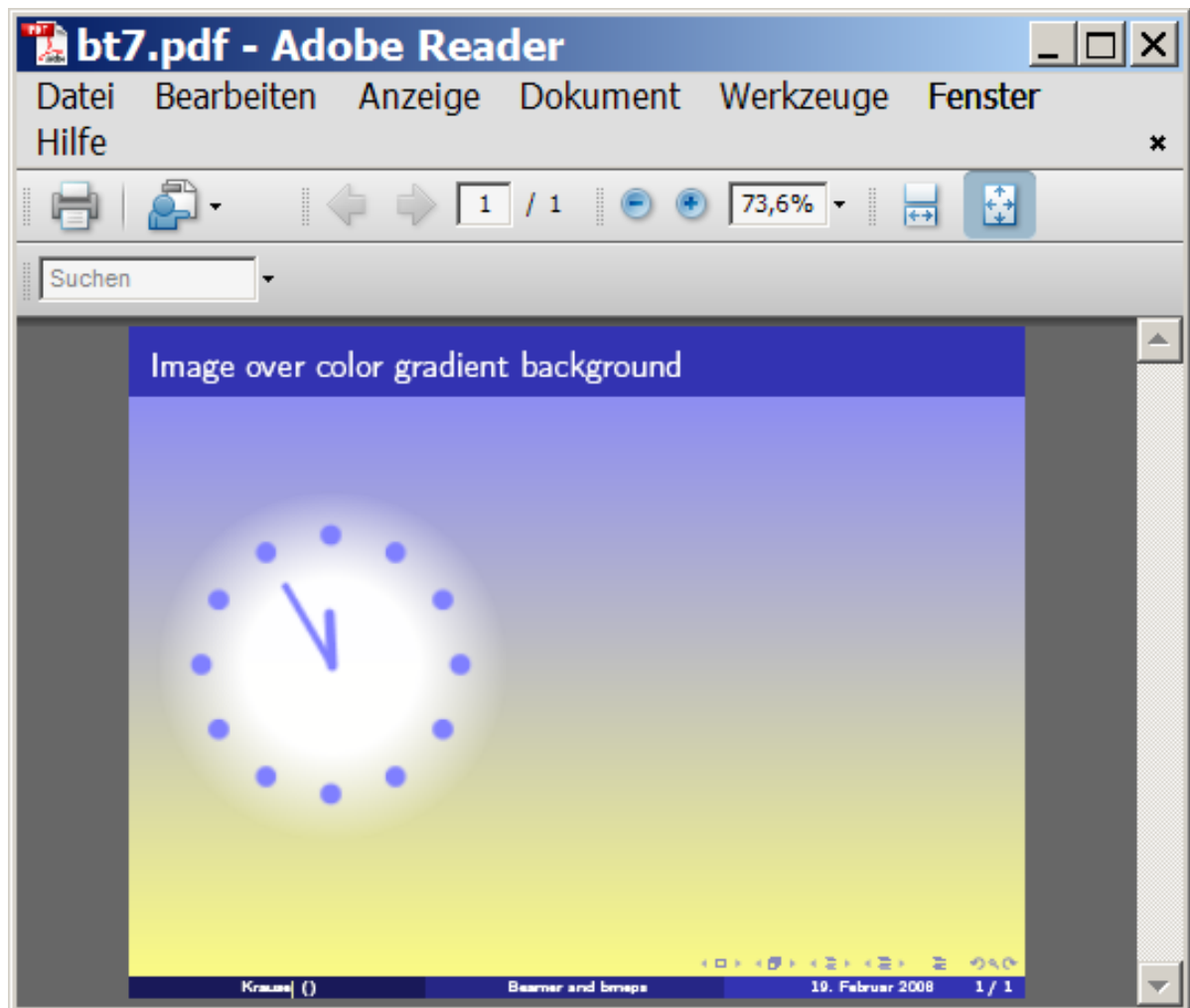


Abbildung 8: Beispiel 6

4 Weitere Informationen zur Verarbeitung verschiedener Bitmap-Dateitypen

4.1 PNG

PNG ist das von bmeips bevorzugte Eingabe-Dateiformat, weil

- PNG verlustfreie Komprimierung mit hoher Kompressionsrate einsetzt und
- es zum Einlesen von PNG-Dateien eine Bibliothek libpng gibt. Diese Bibliothek wird auch als PNG reference library bezeichnet und kann mit allen Bilddateien umgehen, die dem Standard entsprechen.

Das Dateiformat PNG ist besonders geeignet für „künstliche“ Bilder, z.B. am PC erzeugte Graphiken, Logos, Screenshots. . .

4.2 JPEG

Das Dateiformat JPEG verwendet DCT-Komprimierung, ein verlustbehaftetes Kompressionsverfahren mit variabler Kompressionsrate. Es ist besonderes geeignet für Bilder, in denen hauptsächlich fließende Übergänge und Farbverläufe auftreten, z.B. Photos. Für Bilder mit größeren gleichfarbigen Flächen und scharfen Kanten (z.B. Screenshots. . .) ist es weniger geeignet.

JPEG-Dateien, die dem JFIF-Standard und einigen weiteren Anforderungen entsprechen, können 1:1 als DCT-codierte PostScript-Daten verwendet werden, bmeips nimmt lediglich eine zusätzliche ASCII-Hex- bzw. ASCII85-Codierung vor. Für die 1:1-Übertragung müssen alle folgenden Voraussetzungen erfüllt sein:

- Die Analysefunktion von bmeips erkennt die JPEG-Datei als standardgemäß.
- In der JPEG-Datei wird exakt eine Begin-Image-Anweisung gefunden.
- Die JPEG-Datei enthält nur zugelassene SOF_n-Marker.
- Es ist keine Umwandlung einer farbigen Eingabedatei in eine Ausgabedatei mit Graustufen erforderlich.

Sind nicht alle diese Voraussetzungen erfüllt, wird die JPEG-Datei mittels der jpeglib-Bibliothek analysiert und das Image decodiert, die PostScript-Ausgabe erfolgt dann wie bei den anderen Dateitypen auch als Kombination von Run-Length-, Flate- und ASCII-Hex-/ASCII85-Codierung.

4.3 NetPBM

Mit den NetPBM-Tools kann eine Vielzahl von Dateitypen in das NetPBM-Format gewandelt werden, dieses Format wird daher häufig als temporäre Datei in Zwischenschritten bei Graphikumwandlungen erzeugt.

Die NetPBM-Tools können mit `bmeps` mit Hilfe von Pipes kombiniert werden, z.B mit:

```
xxxtopnm input.xxx | bmeps -lpdf -tpnm > output.pdf
```

Nachteilig bei diesem Verfahren ist allerdings, dass ein evtl. in der Eingabedatei vorhandener Alpha-Kanal hier nicht benutzt wird. Wenn die Eingabedatei einen Alpha-Kanal enthält, sollte eine Umwandlung zu einer PNG-Datei erfolgen.

4.4 TIFF

TIFF ist nicht ein Dateiformat sondern bezeichnet eine ganze Gruppe unterschiedlicher Bildformaten mit verschiedenen Kompressions- und Codierungsverfahren. Die Kompressions- und Codierungsverfahren nebst den Bilddaten werden in der Datei als Tags abgespeichert (Datensätze, entspricht in etwa der Bezeichnung Chunk in anderen Dateiformaten), daher die Bezeichnung „tagged image file format“.

Ein Teil der Tags ist unter dem Namen „TIFF 6.0 baseline“ standardisiert, daneben existieren eine Reihe zusätzlicher Tags verschiedener Softwarehersteller. Es ist somit nicht ungewöhnlich, dass Programme bei der Verarbeitung von TIFF-Dateien bestimmte Tags nicht verarbeiten können.

`bmeps` benutzt die Funktion `TIFFReadRGBAImage()` aus der `libtiff`-Bibliothek, um TIFF-Dateien zu lesen. Dies bedeutet, dass nur solche TIFF-Dateien verwendet werden können, die die nachfolgenden Bedingungen erfüllen:

- Die Dateien müssen zum Standard „V 6.0 baseline“ konform sein.
- Die Anzahl der Bits pro Farbkomponente (Rot, Grün, Blau, Alpha) darf 8 nicht überschreiten.
- Die Bildabmessungen dürfen nur so groß sein, dass der erforderliche Speicher $4 \cdot w \cdot h$ Bytes in einen dynamisch allozierbaren Speicherblock passt.

Der TIFF-Support für `bmeps` wurde entwickelt, um Dateien von einem Faxgateway für einfachere Betrachtung in das PDF-Format umzuwandeln. Dies wurde erfolgreich getestet.

Bei der Verwendung von `bmeps` auf andere TIFF-Dateien aus diversen Quellen wurden in der Testphase Fehler und Warnungen beobachtet. Dies ist nicht als Bug in `bmeps` anzusehen, bitte senden Sie mir in solchen Fällen keine Bug-Reports zu.

Fazit: Falls Ihr Zeichenprogramm neben der Ausgabe von TIFF auch die Ausgabe von PNG-Dateien anbietet, sollten Sie PNG bevorzugen.

Ist dies nicht möglich, können Sie auch mit verschiedenen Graphikprogrammen (z.B. Gimp, XnView...) versuchen, die TIFF-Dateien zu öffnen und als PNG zu speichern.

5 Neuerungen in Version 2.0.0

5.1 Robustere PS/EPS-Ausgabe

Vorangegangene Version von bmeps erstellten eine EPS-Ausgabe nach ungefähr folgendem Schema:

```
/pstr 128 string def
/inputf currentfile /ASCII85Decode
filter /RunLengthDecode filter def
gsave
0 128 translate 128 128 scale
128 128 8 [128 0 0 -128 0 0] { inputf pstr readstring pop } image
K)^H&K)^H&K)^H&K)^H&K)^H&K)^H&K)^H&\,QR/mHa_*rrq/DNK'1UM#RM<Nr4k:ied[8
% .... weitere codierte Daten
K)'=[
~>
grestore
currentdict /inputf undef currentdict /pstr undef
```

Nachteilig ist hierbei, dass einige PS-Anweisungen noch nach den komprimierten und codierten Image-Daten stehen. Kommt es bei der Decodierung der Image-Daten zu Fehlern, liest der PS-Interpreter an der Stelle in den codierten Daten weiter, an der der Abbruch erfolgte. Die codierten Daten, die dann gelesen werden, ergeben keine sinnvollen PS-Anweisungen, also wird der Interpreter mit einer Fehlermeldung abbrechen.

Weiterhin werden hier zwar durch die filter-Anweisungen Dateien geöffnet, aber nicht wieder geschlossen.

Die Ausgabe von bmps wurde mit der PS-Ausgabe anderer Programme verglichen, hier ist insbesondere jpeg2ps hervorzuheben. Version 2.0.0 verwendet ein Schema wie folgt:

```
{
gsave
13 dict begin
/fa currentfile /ASCII85Decode filter def
/fb fa /FlateDecode filter def
/sr 128 string def /sg 128 string def /sb 128 string def
/DeviceRGB setcolorspace
0 128 translate 128 128 scale
<<
/ImageType 1
/Width 128 /Height 128 /ImageMatrix [128 0 0 -128 0 0]
/MultipleDataSources true
/DataSource [
{ fb sr readstring pop }
{ fb sg readstring pop }
{ fb sb readstring pop }
]
/BitsPerComponent 8
/Decode [0 1 0 1 0 1]
/Interpolate true
>>
image
fb closefile
fa flushfile fa closefile
end
grestore
} exec
J3Vsg3$]7K#D>EP:q1$o*==mro@So+\<\5,H7Uo<*.jE<[.O@Wn[3@'nb-^757;Rp>H
?K;B=Q#WO;koDO!%M.<jFY8C$]2rn<?/*&:mfh[_=,uiX>KqH^7HqKeXM^,2~>
```

Mit der exec-Anweisung ist dem PS-Interpreter bereits die ganze Prozedur bekannt – einschließlich der Aufräumarbeiten zum Schließen von Dateien und dem „Aufessen“ des codierten Dateistromes – bevor mit der Image-Decodierung begonnen wird. Tritt bei der Decodierung ein Fehler auf oder werden nicht alle Daten aus dem Datenstrom benötigt, werden trotzdem alle Dateien geschlossen und der Datenstrom aufgebraucht, bis das Datenende-Zeichen „~>“ auf der untersten Ebene gefunden wird. Anschließend setzt der PS-Interpreter seine Arbeit normal fort.

5.2 Multiple data sources

Version 2.0.0 bietet die Ausgabe farbiger PS/EPS-Daten mit getrennten Datenquellen an. Ist z.B. eine rote Fläche auszugeben, ergab dies bei vorangegangenen Versionen einen Datenstrom mit den Bytes 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0x00...

Bei getrennten Datenquellen für Rot, Grün und Blau liegen drei Datenströme 0xFF, 0xFF, 0xFF..., 0x00, 0x00, 0x00... und 0x00, 0x00, 0x00... vor. Für diese Datenströme ist eine Run-Length-Komprimierung möglich.

In der PS/EPS-Ausgabe werden die Bilddaten zeilenweise ausgegeben, bei getrennten Datenströmen jeweils erst alle Rot-Daten, dann die Grün-Daten und anschließend die Blau-Daten. Der image-Operator benutzt zum Einlesen Funktionen, die einen String mit Daten füllen und diesen String dann auf den Stack legen. Die Größe von Strings ist begrenzt auf maximal 16384 Bytes.

Fazit: Getrennte Datenquellen können nur verwendet werden, wenn die gepackten Bits für eine Farbe einer Ausgabezeile in einem String untergebracht werden können, der nicht größer als 16383 Bytes ist. Andernfalls wird das Feature „getrennte Datenströme“ automatisch deaktiviert.

5.3 Passthrough von JPEG-Dateien

Vorangegangene Versionen von bmps nutzten immer die jpeglib-Bibliothek, um JPEG-Dateien zu decodieren, anschließend wurden die Bilddaten mittels Kombination von Run-Length-, Flate- und ASCII85- bzw. ASCII-Hex-Codierung in die Ausgabedatei geschrieben.

Dieses Verfahren ist zeitaufwendig und erzeugt in den meisten Fällen Ausgabedateien, die erheblich größer als die Original-JPEG-Dateien sind.

Daher wird ab bmps 2.0.0 bei JPEG-Dateien geprüft, ob eine 1:1-Übernahme der DCT-komprimierten Bilddaten möglich ist, diese Daten werden dann nur mittels ASCII85- oder ASCII-Hex-Codierung nachbehandelt (siehe Abschnitt 4.2 auf Seite 32).

5.4 Flexible Anzahl Bits pro Farbkomponente

Vorangegangene Versionen von bmeps konnten nur PS/EPS mit 8 Bits pro Farbkomponente (bits per component) erzeugen.

Version 2.0.0 nutzt alle in PS/EPS bzw. PDF verfügbaren Farbtiefen, d.h. 1, 2, 4, 8 und 12 Bits pro Farbkomponente für PS/EPS bzw. 1, 2, 4 oder 8 Bits pro Farbkomponente für PDF.

Haben die Eingabedaten eine abweichende Farbtiefe wird die Farbtiefe nach Möglichkeit auf den nächsthöheren möglichen Wert erhöht bzw. auf 12 oder 8 Bits pro Farbkomponente begrenzt.

5.5 PDF-Ausgabe

Vorangegangene Versionen konnten nur PS/EPS-Ausgaben erzeugen, ab Version 2.0.0 erzeugt bmeps auch PDF-Dateien.

Das bereits existierende Programm png2pdf erlaubt zwar die Erzeugung von PDF, allerdings nur aus PNG-Dateien. Bmeps 2.0.0 kann außerdem auch JPEG, NetPbm und teilweise auch TIFF-Dateien nach PDF konvertieren.